

Sensibilité des algorithmes d'ordonnancement

Une étude expérimentale

Jean-Noël Quintin, **Swann Perarnau**

Équipe-Projet MOAIS, Laboratoire d'Informatique de Grenoble/INRIA

RenPar 2011

Imaginons la conception et le déploiement d'un ordonnanceur..

Imaginons la conception et le déploiement d'un ordonnanceur..

- 1 Observation de l'environnement cible,

Imaginons la conception et le déploiement d'un ordonnanceur..

- 1 Observation de l'environnement cible,
- 2 Modélisation,

Imaginons la conception et le déploiement d'un ordonnanceur..

- 1 Observation de l'environnement cible,
- 2 Modélisation,
- 3 Conception d'un algorithme,

Imaginons la conception et le déploiement d'un ordonnanceur..

- 1 Observation de l'environnement cible,
- 2 Modélisation,
- 3 Conception d'un algorithme,
- 4 Validation théorique,

Imaginons la conception et le déploiement d'un ordonnanceur..

- 1 Observation de l'environnement cible,
- 2 Modélisation,
- 3 Conception d'un algorithme,
- 4 Validation théorique,
- 5 Validation expérimentale (simulation),

Imaginons la conception et le déploiement d'un ordonnanceur..

- 1 Observation de l'environnement cible,
- 2 Modélisation,
- 3 Conception d'un algorithme,
- 4 Validation théorique,
- 5 Validation expérimentale (simulation),
- 6 Déploiement.

Problème

Il n'y a pas toujours adéquation entre l'algorithme et l'environnement.

Problème

Il n'y a pas toujours adéquation entre l'algorithme et l'environnement.

À l'origine :

Le modèle peut être plus simple que la véritable plateforme.

Exemple : simplification de la topologie réseau.

Problème

Il n'y a pas toujours adéquation entre l'algorithme et l'environnement.

À l'origine :

Le modèle peut être plus simple que la véritable plateforme.

Exemple : simplification de la topologie réseau.

Dans la durée :

Nouvelles classes d'applications ou nouveaux matériels.

Exemple : Graph500, similaire au Top500 mais avec plus E/S.

Objectif

Sensibilité d'un ordonnanceur :

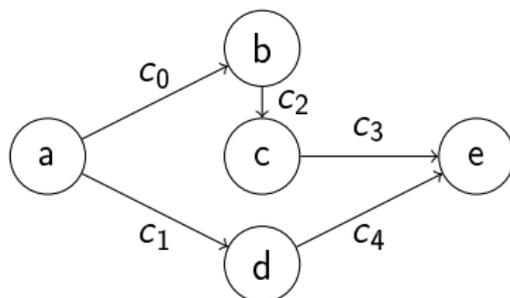
Comment évolue la performance d'un ordonnanceur lorsque l'on modifie son environnement ?

Ici :

Simulation d'un jeu d'ordonnanceurs issus de la littérature.
Modifications des applications en entrée.

Ordonnanceurs étudiés

Tâches avec précédences, coûts de communication.



Entrées : Graphes orientés acycliques

- Les noeuds représentent le calcul.
- Les arêtes les communications.
- Annotations supplémentaires pour les coûts.

Plan

- 1 Protocole Expérimental
- 2 Analyse de sensibilité
 - Performance de référence
 - Coûts en calcul
 - Coûts en communication
- 3 Conclusion

Plan

- 1 Protocole Expérimental
- 2 Analyse de sensibilité
 - Performance de référence
 - Coûts en calcul
 - Coûts en communication
- 3 Conclusion

Schéma des expériences

- 1 Définition de jeux de graphes de référence.
- 2 Génération aléatoire des graphes.
- 3 Annotation des graphes selon des distributions uniformes.
- 4 Simulation des algorithmes étudiés → performance de référence.
- 5 Annotation des graphes avec une autre distribution.
- 6 Nouvelles simulations → performance perturbée.
- 7 Comparaison des performances pour chaque ordonnanceur.

Outils utilisés

GGen

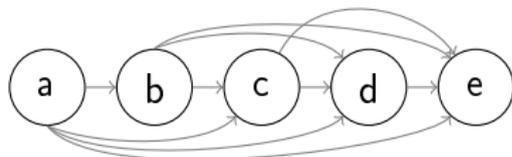
Outil de génération de graphes pour l'ordonnancement.
Implémentation de qualité des algorithmes les plus classiques.
Séparation de la génération du graphe de son annotation.
Sorties en DOT (Graphviz).

SimGrid

Outil de simulation pour la grille.
Implémentation de quelques algorithmes d'ordonnancement.
Spécification détaillée de la plateforme d'exécution.

Layer-by-Layer [Kasahara *et al.*, 2002]

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | X | - | - | - | - |
| b | X | X | - | - | - |
| c | X | X | X | - | - |
| d | X | X | X | X | - |
| e | X | X | X | X | X |



Paramètres

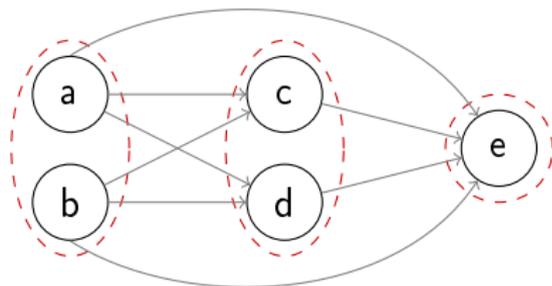
n : Nombre de nœuds.

l : Nombre de niveaux.

p : Probabilité d'apparition d'une arête.

Layer-by-Layer [Kasahara *et al.*, 2002]

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | X | - | - | - | - |
| b | X | X | - | - | - |
| c | X | X | X | - | - |
| d | X | X | X | X | - |
| e | X | X | X | X | X |



Paramètres

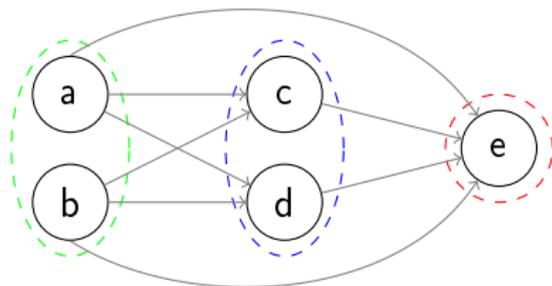
n : Nombre de nœuds.

l : Nombre de niveaux.

p : Probabilité d'apparition d'une arête.

Layer-by-Layer [Kasahara *et al.*, 2002]

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | X | X | - | - | - |
| b | X | X | - | - | - |
| c | X | X | X | X | - |
| d | X | X | X | X | - |
| e | X | X | X | X | X |



Paramètres

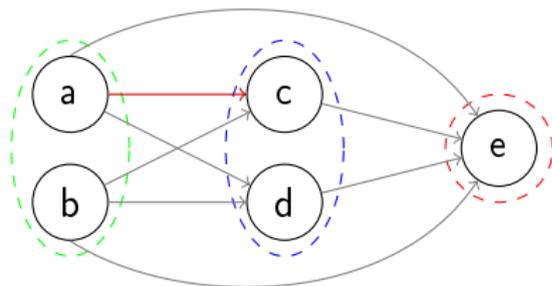
n : Nombre de nœuds.

l : Nombre de niveaux.

p : Probabilité d'apparition d'une arête.

Layer-by-Layer [Kasahara *et al.*, 2002]

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | X | X | - | - | - |
| b | X | X | - | - | - |
| c | X | X | X | X | - |
| d | X | X | X | X | - |
| e | X | X | X | X | X |



Paramètres

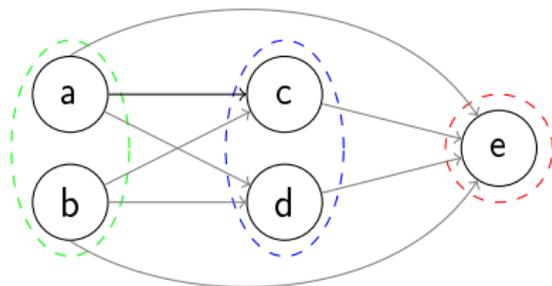
n : Nombre de nœuds.

l : Nombre de niveaux.

p : Probabilité d'apparition d'une arête.

Layer-by-Layer [Kasahara *et al.*, 2002]

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | X | X | 1 | - | - |
| b | X | X | - | - | - |
| c | X | X | X | X | - |
| d | X | X | X | X | - |
| e | X | X | X | X | X |



Paramètres

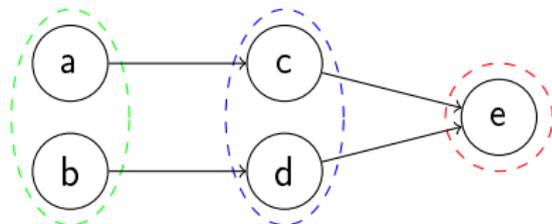
n : Nombre de nœuds.

l : Nombre de niveaux.

p : Probabilité d'apparition d'une arête.

Layer-by-Layer [Kasahara *et al.*, 2002]

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | X | X | 1 | 0 | 0 |
| b | X | X | 0 | 1 | 0 |
| c | X | X | X | X | 1 |
| d | X | X | X | X | 1 |
| e | X | X | X | X | X |



Paramètres

n : Nombre de nœuds.

l : Nombre de niveaux.

p : Probabilité d'apparition d'une arête.

Algorithmes choisis

HEFT

Par liste : *BottomLevel*.

CPOP

Par liste : longueur du chemin critique.

DSC

Clustering : fusions des tâches avec les plus grandes communications.

HBMCT

Hybride : liste par niveau, regroupement par vitesse d'exécution.

Jeux de données témoins

| Entrées | Nb tâches | Nb Comm. | | Coût en calcul | | Coût en Comm. | | CCR |
|-------------|-----------|----------|------------|----------------|------------|---------------|------------|--------------|
| | | moy | écart-type | moy | écart-type | moy | écart-type | |
| T_{small} | 500 | 746 | 27 | 9,98 | 5.7 | 0.5 | 0.2 | ≈ 20 |
| T_{moy} | | | | | | 5.1 | 2.5 | ≈ 2 |
| T_{big} | | | | | | 10.2 | 5.1 | ≈ 1 |

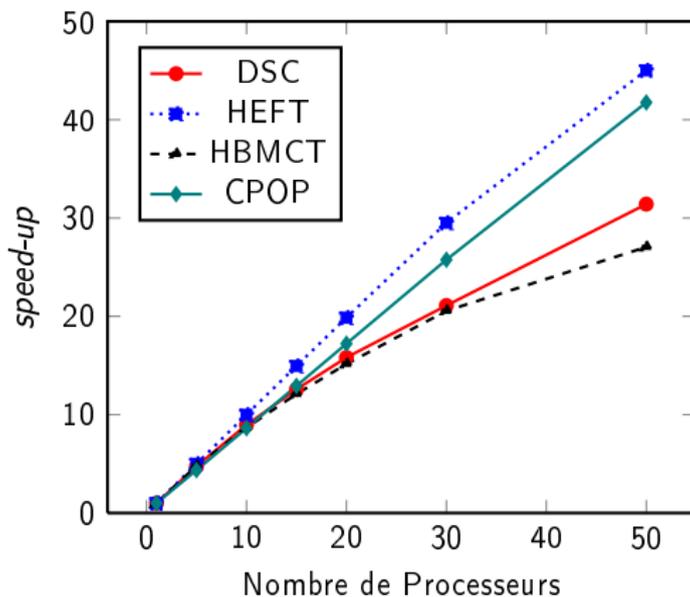
Table: Récapitulatif des caractéristiques des jeux de données de référence. Les coûts sont donnés en moyenne sur une tâche/arête.

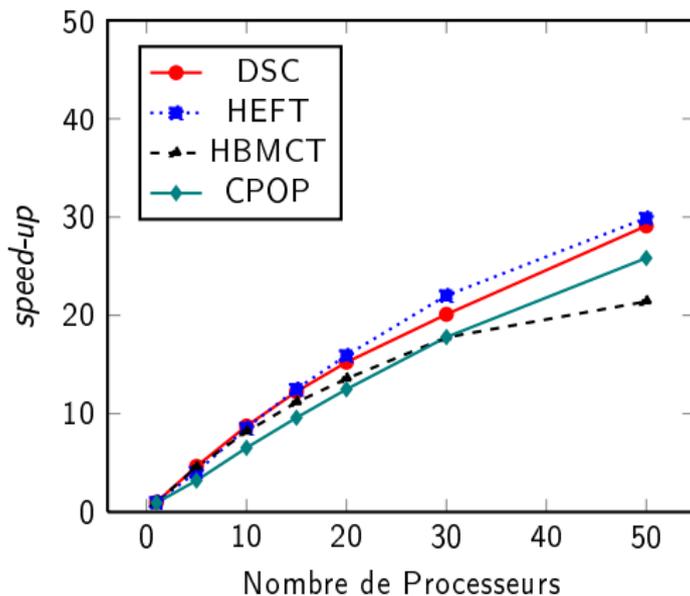
Paramètres de la simulation

- Nombre variable de processeurs.
- Topologie réseau en clique.
- 1000 simulations par point.

Plan

- 1 Protocole Expérimental
- 2 Analyse de sensibilité
 - Performance de référence
 - Coûts en calcul
 - Coûts en communication
- 3 Conclusion

Performance de référence (T_{small})

Performance de référence (T_{big})

Étude de sensibilité

Modifications réalisées :

Mêmes graphes (structure) que les jeux témoins.

Modification des annotations : distribution exponentielle.

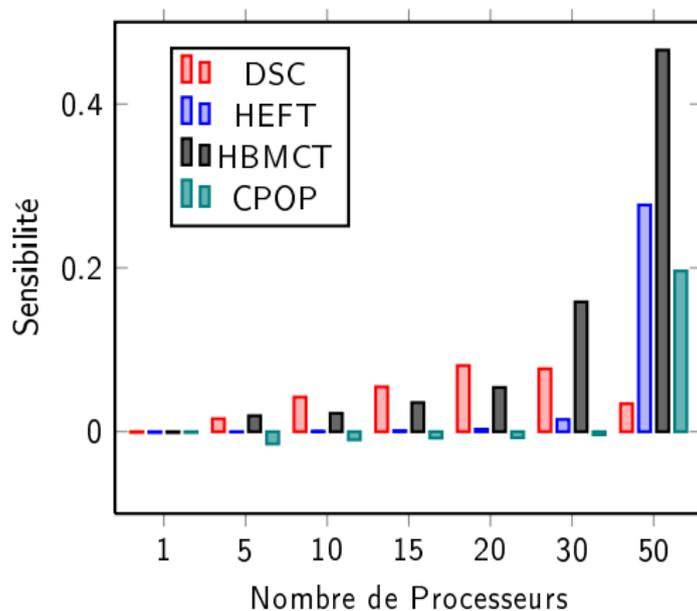
Conservation de la moyenne.

Sensibilité = $\frac{C_{max} - C_{max_ref}}{C_{max_ref}}$ avec :

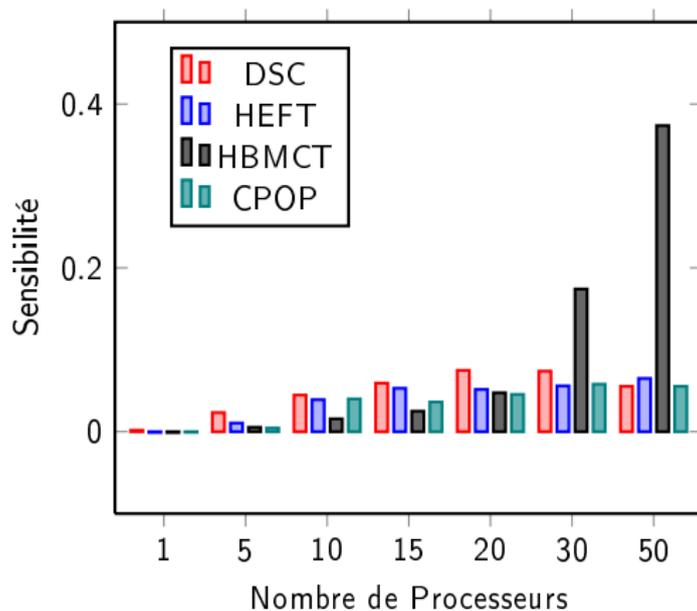
- C_{max} : le temps d'exécution sur jeu d'entrées modifiées.
- C_{max_ref} : le temps d'exécution obtenu sur le groupe test.

Plan

- 1 Protocole Expérimental
- 2 Analyse de sensibilité
 - Performance de référence
 - Coûts en calcul
 - Coûts en communication
- 3 Conclusion

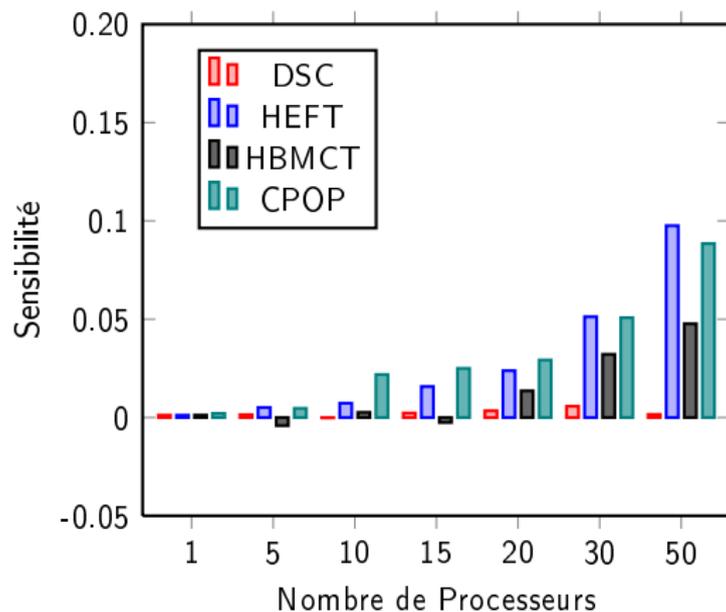
Sensibilité aux calculs (T_{small})

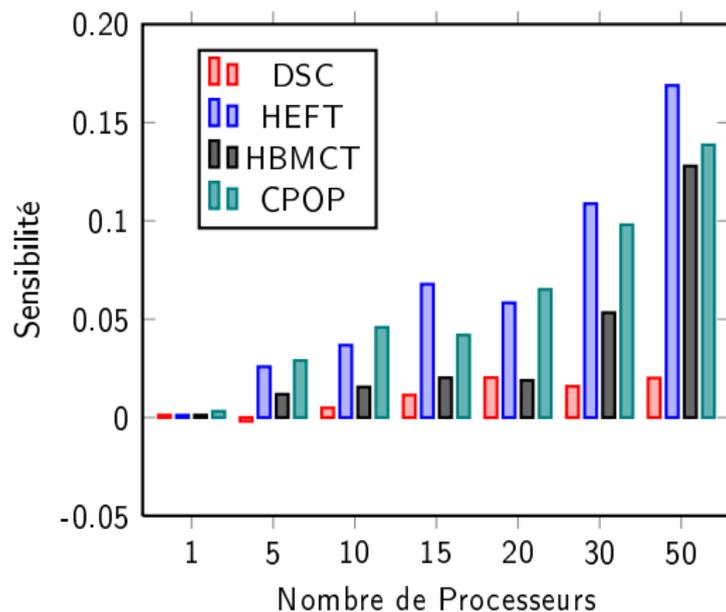
Sensibilité aux calculs (T_{big})



Plan

- 1 Protocole Expérimental
- 2 Analyse de sensibilité
 - Performance de référence
 - Coûts en calcul
 - Coûts en communication
- 3 Conclusion

Sensibilité aux communications (T_{moy})

Sensibilité aux communications (T_{big})

Inversions

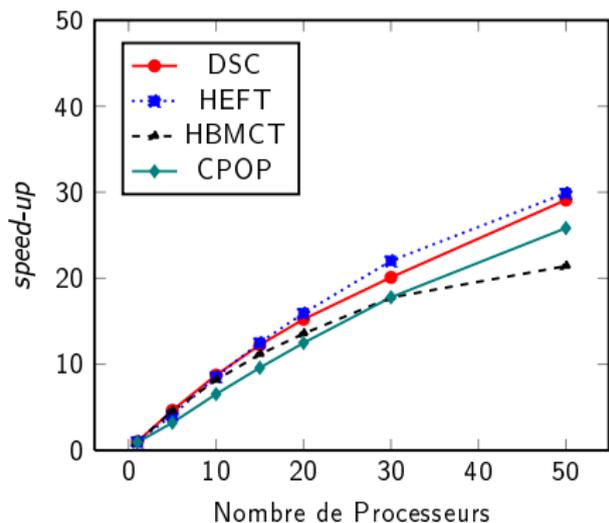


Figure: T_{big}

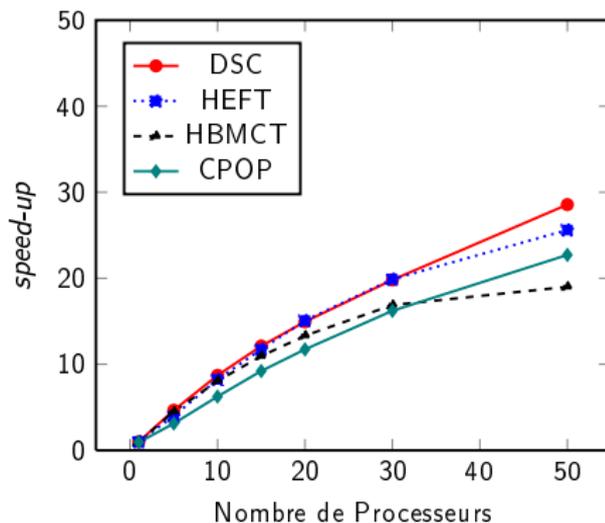


Figure: T_{big} modifié

Plan

- 1 Protocole Expérimental
- 2 Analyse de sensibilité
 - Performance de référence
 - Coûts en calcul
 - Coûts en communication
- 3 Conclusion

Conclusion

Une première étude de sensibilité des ordonnanceurs :

Campagne expérimentale à grande échelle.

Utilisation d'un simulateur éprouvé.

Génération des données aléatoires par des méthodes reconnues.

Tests de différentes classes d'algorithmes.

Identification d'inversions.

Chaque algorithme réagit différemment à la même modification !

Perspectives

Nouvelles Simulations

Différentes méthodes de génération.

Utilisation d'autres distributions aléatoires.

Modification des caractéristiques de la plateforme.

Nouvelles Analyses

Comment exprimer plus formellement les perturbations des entrées ?

Comment explorer/détecter ces sensibilités et les inversions ?

Merci de votre attention.